CLAIMS

1. A controller for a random access memory comprises:

an address and command queue that holds memory references from a plurality of microcontrol functional units, said address and command queue comprising:

a read queue;

a first read/write queue that holds memory references from a core processor; and

control logic including an arbiter that detects the fullness of each of the queues and a status of completion of outstanding memory references to select a memory reference from one of the queues.

2. The controller of claim 1 wherein the control logic further selects one of the queues to provide a next memory references based on a programmable value stored in a priority service control register.

3. The controller of claim 1 wherein the address and command queue comprises:

a high priority queue that holds memory references from high priority tasks.

4. The controller of claim 1 wherein a microengine sorts memory references into read and write memory references.

5. The controller of claim 1 wherein the address and command queue comprises:

an order queue that holds write memory requests wherein the controller examines incoming reference requests and sorts the

incoming memory reference requests into either the read queue or an order queue in accordance with a specified address mode.

6. The controller of claim 5 wherein the address and command queue comprises:

an order queue; and

wherein if the memory reference request does not have a memory optimization bit set, the memory reference is stored in the order queue.

7. The controller of claim 1 wherein the address and command queue is implemented in a single memory structure and comprises:

an order queue for storing memory references;

an read queue for storing memory references;

a high priority queue for storing memory references; and

with the memory structure being segmented into four different queue regions, each region having its own head and tail pointer.

8. The controller of claim 7 wherein the address and command queue further comprises:

an insert queue control and a remove queue arbitration logic to control insert and removal of memory references from the queues.

9. The controller of claim 1 further comprising:

a command controller and address generator that is responsive to an address from a selected memory reference from one said queues, to produce addresses and commands to control a

memory interface.

10. The controller of claim 9 further comprising:

a memory interface responsive to generated addresses and commands to produce memory control signals.

11. The controller of claim 9 wherein controller further comprises:

a lock lookup content addressable memory for look-ups of read locks.

12. The controller of claim 10 wherein the address and command queue further comprises:

a Read Lock Fail Queue to hold read lock memory reference requests that fail because of a preexisting lock is on a portion of memory.

13. The controller of claim 12 wherein controller comprises control logic to respond if one of the microengines issues a read lock request by accessing the lock lookup memory to determine whether the memory location is already locked.

14. The controller of claim 13 wherein if the memory location is locked from any prior read lock request, then the issued memory lock request fails and is stored in the read lock fail queue.

15. The controller of claim 14 wherein if the memory location is not locked then the issued memory reference is converted into address signals for the memory by the memory interface.

16. The controller of claim 15 wherein the command controller and address generator enters the lock for the memory address for the issued memory reference into the lock look up memory.

17. The controller of claim 1 wherein the controller is configured to control static random access memory (SRAM).